PowerEnJoy

Politecnico di Milano

Industrial and Information Engineering Computer Science and Engineering

Cattaneo Davide

El Hariry Matteo

Frontino Francesco

Project Plan Document

Contents

1 Introduction

1.1 Purpose and scope 3
1.2 Definitions, acronyms and abbreviations 3
1.3 Reference documents 4

2 Project size, cost and effort estimation

2.1 Overview 5
2.2 Size estimation: function points 5
2.2.1 Internal logic files 7
2.2.2 External logic files 8
2.2.3 External inputs 9
2.2.4 External inquiries 12
2.2.5 External outputs 13
2.2.6 Overall estimation 14
2.3 Costs and effort estimation: COCOMO II 15
2.3.1 Scale drivers 15
2.3.2 Cost drivers 17
2.3.3 Effort equation 29
2.3.4 Schedule estimation 30

3 Schedule 31

4 Resource allocation 33

5 Risk management 35

6 Revision 36

6.1 Team work 36

1 Introduction

1.1 Purpose and scope

Purpose of this document is to give details about how the project is managed during its development cycle. Function points and COCOMO costs estimation methodologies are taken into account to generate the tables reported in chapter 2.

1.2 Definitions, acronyms and abbreviations

Here is a brief description of the most important actors and words used in our system:

- FP: Function Points
- ILF: Internal logic file
- ELF: External logic file
- EI: External Input
- EO: External Output.
- EQ: External Inquiries
- DBMS: Database Management System.
- API: Application Programming Interface

1.3 Reference documents

Specification Document:
* Assignments 1 and 2 (RASD and DD).pdf
* ITPD.pdf

Examples documents:
* Project plan example document.pdf

2 Integration strategy

2.1 Overview

This section is specifically focused on providing some estimations of the expected size, cost and required effort of PowerEnjoy.
For the size estimation part we will essentially use the Function Points approach, taking into account all the main functionalities of PowerEnjoy and estimating the correspondent amount of lines of code to be written in Java. This estimation will only take into account the parts of the project that concur to the implementation of the business logic and will disregardthe aspects concerning the user interface.
For the cost and effort estimation we will instead rely on the COCOMO approach, using as initial guidance the amount of lines of code computed with the FP approach.

2.2 Size estimation: function points

A function point is a "unit of measurement" to express the amount of business functionality an information system (as a product) provides to a user. Function points are used to compute a functional size measurement (FSM) of software.
An important assumption is that the dimension of PowerEnJoy platform is characterized based on the functionalities that it has to offer.
The Function Points approach provides an estimation of the size of a project taking as inputs the amount of functionalities to be developed and their complexity.
The estimation is based on the usage of figures obtained through statistical analysis of real projects, which have been properly normalized and condensed in the following tables:

For Internal Logic Files and External Logic Files:

	Data Elements		
Record Elements	*1-19*	*20-50*	*51+*
1	Low	Low	Avg
2-5	Low	Avg	High
6+	Avg	High	High

For External Output and External Inquiry:

	Data Elements		
File Types	*1-5*	*6-19*	*20+*
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

For External Input:

	Data Elements		
File Types	*1-4*	*5-15*	*16+*
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

UFP Complexity Weights:

	Complexity Weight		
Function Type	*Low*	*Average*	*High*
Internal Logic Files	7	10	15
External Logic Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

2.2.1 Internal logic files (ILFs)

This paragraph is going to show what kind of information PowerEnJoy has to store in its internal system in order to offer all the functionalities that will be furthermore explained.
Here is the list of ILFs the system must organize:

- Login data: the login details of the system are saved in two different cases of users indicated by a flag: System users and staff. For staff is an additional flag that differentiates the administrators by the operators. In this table in addition to the flags are posted username and password to enter.
- User data: user data is SSN, name, surname, date of birth, credit card number with security code, license number.
- Car info and status: The data being stored for each electric car are: an id (which is assigned by the system whenever a new vehicle is added to the system) and the license plate. Each car has a status which contains information concerning battery level, position (given by geographical coordinates of latitude and longitude).
- Ride and Reservations: For each ride are saved hours of booking the car start and stops (if it consists of more flights for each start and end time and cost saving intermediate), a progressive positions list, saved as route that trace the path of travel, the user who made the race.

  Access permission info: to keep track of who has access to the administration system services, the staff account information is stored in a table. The main fields are first name, last name, user name, password and id of the privilege level of the account, which is related to the flag indicating if the account belongs to an administrator or an operator, which refers to a corresponding entry in a table privileges.
- API permissions: Finally, the system maintains a list of applications and plugins identifiers which have privileged access to the API. Each identifier is associated with the administrator contact information, a description of what the application or plugin and a list of methods that can be called.

Using the previously defined tables, this is the ILF functions points table obtained:

ILF	Complexity	FPs
Login data	Low	7
User data	Low	7
Car info	Low	7
Car status	Low	7
Ride and Reservations	Average	10
API permissions	Average	10
Total		48

2.2.2 External logic files (ELFs)

PowerEnJoy relies, for the collection of external information, on the Maps Service and on the Payment Gateway.
Regarding the Payment Gateway, the only interaction is given by the request for execution of a payment and the reception of the transaction outcome.

Concerning the Map Service, interactions are given by:

- request for location of coordinates of a provided address
- request of possible routes and estimated travel time from one location to another
- provide availability of a geographic representation of the area around a given position.
  This is the information that maps must manage to provide the service used by user app, by the central system and by the staff browser.

The interaction between the core system and the remote service provider happens through a RESTful API and data can be returned in JSON or XML format. The results have then to be processed before they can be used as part of system's computation.
Given the complexity of the interaction and the amount of data that is retrieved during the maps request and responses interaction, it is reasonable to classify the Maps Service logic file as complex. The Payment

Gateway files though, are considered as simple ones, because of the trivial information exchange.

ELF	Complexity	FPs
ETA computation	Low	10
Reverse geocoding	Low	10
Map data retrieval	Low	10
Request to Payment Gateway	Low	10
Payment Gateway response	Low	10
Total		50

2.2.3 External inputs (EIs)

External inputs that PowerEnJoy system receives are heterogeneous given the multiplicity and of different sources.

The inputs can come from web user interfaces, user apps, staff browsers and vehicle's Board Controller.

Users App and staff Browsers:

- Login/Logout: these are simple operations that involve only the account manager. They contribute 3 FPs each.
- Search Car location: this is an operation of average complexity since it only concerns the retrieval of a set of coordinates, 6FPs are then assigned to it.

User App and User Web Interface:

- Register a new account: this operation also has an average complexity, as it involves a number of checks on the validity of the fields. As such, it contributes 4 FPs.

User App:

- Password retrieval: this operation has an average complexity, as it involves a number of steps in order to be sure the user is really entitled to retrieve his password. For this reason, it contributes 4 FPs.
- Change settings: this operation also has an average complexity, as the number of settings to be managed can be quite high. As such, it contributes 4 FPs.
- Reserve a car: this is a complex operation that involves a large number of components. For this reason they account for 6 FPs each.
- View profile information: since this is straightforward operation, it yields 3 FPs.

Staff browser:

Administrator side:

- Insert, delete and update Safe Areas: these are very complex operations that involve a great number of components. For this reason they account for 6 FPs each.
- Insert, delete and update Car and Car's status: as for the safe areas, the complexity of these operations is high, so they account for another 6 FPs each.
- Request service statistics: as this operation involves some fairly complicated aggregate queries on the database, it can be considered complex and thus contributes 6FPs to the total amount.
- Grant and revoke privileges to an application or plugin: while these two specular operations involve a large number of fields that can be set, the impact on the database is quite limited. For this reason we think they have an average complexity and they should contribute 4 FPs each.

- Assign task to Operator: since this operation consist simply in sending a message it has average complexity and it should contribute 4 FPs.

Operator side:

- Send/receive messages: simply in sending a message it has average complexity and it should contribute 4 FPs.
- Create report: Easy operation that implies an append of a text/img report into the DB, so it is assigned 3FPs.
- Update Car Status: a low complexity operation, that updates an instance related to the car status, either in the running class object or both in the object and the database. 3FPs.

Board Controller:

- Send status info: it involves few components that just need to forward a set of variables, 3 FPs.
- Send Ride information: it involves more components that need to forward a large set of variables, 4 FPs.

The final results are shown in the following table:

EI	Complexity	FPs
Login/Logout	Low	2x3
Password retrieval	Average	4
Change settings	Average	4
Reserve a car	High	2x4
View profile info	Low	3
Register a new account	Average	4
View reservation history	Low	3
Insert, delete and update safe areas	High	3x6
Insert, delete and update cars/car status	High	3x6
Request service statistics	High	6
Grant and revoke app privileges	Average	2x4

Grant and revoke plugin privileges	Average	2x4
Search Car location	Average	6
Send status info	Low	3
Send/receive messages	High	3x6
Create report	Low	3
Update Car Status	Low	3
Send Ride information	Average	4
Total		127

2.2.4 External inquiries (EQs)

An inquiry is a data retrieval request performed by an agent of the system.

PowerEnJoy ensures that the following information can be inquired according to the agent who makes the request, as it is described here:

- A user can retrieve his reservation history.
- An administrator can retrieve the full list of users, safe areas, cars, cars status log, and rides information.
- An operator can retrieve cars status log and users list.

All these operations can be though as fairly simple.

The resulting table is the following:

EQ	Complexity	FPs
Retrieve users reservation history	Low	3
Retrieve list of users	Low	3
Retrieve list of safe areas	Low	3
Retrieve list of cars	Low	3
Retrieve cars status log	Low	3
Retrieve rides information	Low	3
Total		18

2.2.5 External outputs (EOs)

The main kind of information that PowerEnJoy system must deliver to its agents, in order to provide efficient functionalities, are notifications that must be forwarded to users, administrators, operators and to cars Board Controller:

- Notify the Board Controller that the car has been reserved.
- Notify a user that his request has been accepted.
- Notify a user that his request has been dropped because reservation expired.
- Notify user of payment outcomes.
- Notify operator of new task assignment.
- Notify admin of emergency occurrence.

All these operations can be though as fairly simple.

The resulting table is the following:

EO	Complexity	FPs
Board Controller reservation request notification	Low	4
Request accepted notification	Low	4
Request dropped notification	Low	4
Operator task assignment notification	Low	4
Admin emergency notification	Low	4
Total		20

2.2.6 Overall estimation

The following table summarizes the results of our estimation activity:

Function Type	Value
Internal Logic Files	48
External Logic Files	50
External Inputs	127
External Inquiries	18
External Outputs	20
Total	263

Considering Java Enterprise Edition as a development platform and disregarding the aspects concerning the implementation of the mobile applications (which can be thought as pure presentation with no business logic), we can estimate the total number of lines of code.

Depending on the conversion rate, we have a lower bound of:

$$SLOC = 263 * 46 = 12098$$

and an upper bound of

$$SLOC = 263 * 67 = 16569$$

2.3 Cost and effort estimation COCOMO II

In this section we're going to use the COCOMO II approach to estimate the cost and effort needed to develop PowerEnjoy.

The effort estimation is achieved through a complex, non linear model that takes into account the characteristics of product and process:

PM = A x SizeE x Π1<=i<=nEMi

2.3.1 Scale drivers

The selection of scale drivers is based on the rationale that they are a significant source of exponential variation on a project's effort or productivity variation.

Each scale driver has a range of rating levels, from Very Low to Extra High. Each rating level has a weight, W, and the specific value of the weight is called a scale factor.

In order to evaluate the values of the scale drivers, we refer to the following official COCOMO II table:

Scale Factors	*Very Low*	*Low*	*Nominal*	*High*	*Very High*	*Extra High*
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SFj	6.20	4.96	3.72	2.48	2.48	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SFj	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
SFj	7.07	5.65	4.24	2.83	1.41	0.00

TEAM SFj	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative 1.10	seamless interactions 0.00
PMAT SFj	Level 1 Lower 7.80	Level 1 Upper 6.24	Level 2 4.68	Level 3 3.12	Level 4 1.56	Level 5 0.00

Precedentedness: it reflects the previous experience of our team with the development of large scale projects; If a product is similar to several previously developed projects, then the precedentedness is high.

Since we are not expert in the field, this value will be low.

Development flexibility: Expresses the degree of conformance to software requirements and external interface standards :as the degree of flexibility in the development process with respect to the external specification and requirements. In this project all the functional requirements are well specified but there are also some free. Consequently the value assigned is nominal.

Risk resolution: reflects the level of awareness and reactiveness with respect to risks. Since we have a good risk management plan, clear definition of budget and schedule, focus on architectural definition we set the very high.

Team cohesion: Captures the consistency of stakeholder objectives and the willingness of all parties to work together as a team. For our team, the value is very high.

Process maturity: Maturity of the software process used to produce the product; all the goals have been successfully achieved and the software process was followed well, this value is set to level 3.

Scale Driver	*Factor*	*Value*
Precedentedness(PREC)	Low	4.96
Development flexibility (FLEX)	Nominal	3.04
Risk resolution (RESL)	Very high	1.41
Team cohesion (TEAM)	Very high	1.10
Process maturity (PMAT)	Level 3	3.12
Total		**13.63**

2.3.2 Cost drivers

RELY Cost Drivers						
RELY Descriptors	Slightly inconvenience	Easily recoverable losses	Moderate recoverable losses	High financial loss	Risk to human life	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	0.82	0.92	1.00	1.10	1.26	n/a

Required Software Reliability: Degree of assurance to which the software will performs its intended function:

If the effect of a software failure is only slight inconvenience then RELY is very low.

If a failure would risk human life then RELY is very high.

In PowerEnjoy the rely cost is set to nominal since a software failure not cause large financial loss

DATA Cost Drivers						
DATA Descriptors		D/P < 10	10 ≤ D/P ≥ 100	10 ≤ D/P ≥ 1000	D/P > 1000	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	n/a	0.90	1.00	1.14	1.28	n/a

Database size: this cost driver attempts to capture the effect large test data requirements have on product development.

We don't have the ultimate answer, but our estimation given the tables and _elds we have is to reach a 3GB database. Since it is distributed over 10.000-15.000 SLOC, the ratio D/P (measured as testing DB bytes/program SLOC) is between 209 and 314, resulting in the DATA cost driver being high

CPLX Cost Driver						
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	0.73	0.87	1.00	1.17	1.34	1.74

Product Complexity: is set to high since we manage queue and stack control, homogeneous distributed processing and matrix/vector operations.

RUSE Cost Driver						
RUSE Descriptors		None	Across project	Across program	Across product line	Across multiple product lines
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	n/a	0.95	1.00	1.07	1.15	1.24

Required reusability: accounts for the additional effort needed to construct components intended for reuse on the current or future projects.

In our case, the reusability requirements are limited in scope to the project itself, so the RUSE cost driver is set to nominal.

DOCU Cost Driver						
DOCU Descriptors	Many life-cycle needs uncovered	Some life-cycle needs uncovered	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very-excessive for life-cycle needs	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	0.81	0.91	1.00	1.11	1.23	n/a

Documentation match to life-cycle needs: evaluated in terms of the suitability of the project's documentation to its life-cycle needs. In our case, the DOCU cost driver is set to nominal.

TIME Cost Driver						
TIME Descriptors			≤50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	n/a	n/a	1.00	1.11	1.29	1.63

Execution time constraint: measure of the execution time constraint imposed upon a software systems, PowerEnjoy uses a lot of cpu for handle the various requests and for manage and keep alive the whole system. We set this parameter high.

STOR Cost Driver						
STOR Descriptors			≤50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	n/a	n/a	1.00	1.05	1.17	1.46

Storage Constraint: degree of main storage constraint imposed on a software system. As current disk drives can easily contain several terabytes of storage, since we use less than 50%, this value is set to nominal.

PVOL Cost Driver						
PVOL Descriptors		Major change every 12 mo., minor change every 1 mo.	Major: 6mo; minor: 2wk.	Major:2mo , minor:1wk	Major: 2wk; mi-nor: 2 days	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	n/a	0.87	1.00	1.15	1.30	n/a

Platform Volatility: this rating ranges from low, where there is a major change every 12 months, to very high, where there is a major change every two weeks.

This parameter is set nominal. Indeed only the mobile app requires sporadically updates for the major OS releases.

ACAP Cost Driver						
ACAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.42	1.19	1.00	0.85	0.71	n/a

Analyst Capability: the major attributes that should be considered in this rating are Analysis and Design ability, efficiency and thoroughness, and the ability to communicate and cooperate.

The rating should not consider the level of experience of the analyst;

that is rated with AEXP.

Analysts that fall in the 15th percentile are rated very low and those that fall in the 95th percentile are rated as very high.

based on this index, and with respect to how the our analysis was conducted we assign the value high.

PCAP Cost Driver						
PCAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.34	1.15	1.00	0.88	0.76	n/a

Programmer Capability: since in this index we evaluate the capability of the programmers as a team rather than as individuals we set this index to high.

APEX Cost Driver						
APEX Descriptors	≤ 2 months	6 months	1 year	3 year	6 year	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.22	1.10	1.00	0.88	0.81	n/a

Applications Experience: dependent on the level of applications experience of the project team developing the software system or subsystem; We have some experience in the development of Java applications, but we never tackled a Java EE system of this kind. For this

reason we're going to set this parameter to low.

PLEX Cost Driver						
PLEX Descriptors	≤ 2 months	6 months	1 year	3 year	6 year	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.19	1.09	1.00	0.91	0.85	n/a

Platform Experience: the rating for this cost driver (formerly labeled AEXP) is dependent on the level of applications experience of the project team developing the software system or subsystem.

We have some previous experience with databases, user interfaces and server side development. For this reason, we're going to set this parameter to nominal.

LTEX Cost Driver						
LTEX Descriptors	≤ 2 months	6 months	1 year	3 year	6 year	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.20	1.09	1.00	0.91	0.84	n/a

Language and Tool Experience: we don't have any experience with the Java EE platform and with the development of mobile application but we have some previous experience with databases, user interfaces and server side development. We are also knowledgable of the development environment, so we're going to set this parameter to nominal.

Project factors account for influences on the estimated effort such as use of modern software tools, location of the development team, and compression of the project schedule.

PCON Cost Driver						
PCON Descriptors	48% / year	24% / year	12% / year	6% / year	3% / year	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.29	1.12	1.00	0.90	0.81	n/a

Personnel continuity: the rating scale for PCON is in terms of the project's annual personnel turnover: This parameter is set to very high since in our case the act of replacing an employee with a new employee is rare.

TOOL Cost Driver						
TOOL Descriptors	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.17	1.09	1.00	0.90	0.78	n/a

Usage of Software Tools: the tool rating ranges from simple edit and code, very low, to integrated life-cycle management tools, very high.

Our application environment is complete and well integrated, so we'll set this parameter as high.

SITE Cost Driver						
SITE Collocation Descriptors SITE Communications Descriptors	Intern-Ational Some phone, mail	Multi-city and multi-company Individual phone, fax	Multi-city or multi-company Narrow band email	Same city or metro area Wideband electronic communi-cation	Same building or complex Wideband elect. Comm., occasional video conf.	Fully collocated Interactive multimedi a
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.22	1.09	1.00	0.93	0.86	0.80

Multisite development: the team's components lives in different cities and collaborates using different wideband internet services, including e-mail and video conferences. Therefore the value is set to Very High.

SCED Cost Driver						
SCED Collocation Descriptors	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.43	1.14	1.00	1.00	1.00	n/a

Required development schedule:

This rating measures the schedule constraint imposed on the project team developing the software. The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a project requiring a given amount of effort.

Even if the entire work was well distributed over the available development time, the definition of all the required documentation took a consistent amount of time, especially for the requirement analysis and the design phases. For this reason, this parameter is set to high.

Cost Driver	*Factor*	*Value*
Required Software Reliability (RELY)	Nominal	1.00
Database size (DATA)	High	1.14
Product complexity (CPLX)	High	1.17
Required Reusability (RUSE)	Nominal	1.00
Documentation match to life-cycle needs(DOCU)	Nominal	1.00
Execution Time Constraint (TIME)	High	1.11
Main storage constraint (STOR)	Nominal	1.00
Platform volatility (PVOL)	Nominal	1.00
Analyst capability (ACAP)	High	0.85
Programmer capability (PCAP)	High	0.88
Application Experience (APEX)	Low	1.10
Platform Experience (PLEX)	Nominal	1.00
Language and Tool Experience (LTEX)	Nominal	1.00
Personnel continuity (PCON)	Very High	0.81
Usage of Software Tools (TOOL)	High	0.90
Multisite development (SITE)	Very High	0.86
Required development schedule (SCED)	High	1.00
Total		**0,76372**

2.2.3 Effort equation

Effort is expressed in terms of person-months (PM). A person-month is the amount of time one person spends working on the software development project for one month.
COCOMO II defines the following formula to estimate it:

$$\text{Effort} = A * EAF * \text{KSLOC}^{E}$$

Exponent E and parameter EAF and constant A are defined as follows:

$$A = 2.94$$

EAF = Effort Adjustment Factor the product of all cost drivers
$$EAF = \prod_i^n Em_i = 0.76372$$

E = exponent derived from the scale drivers
$$E = B + 0.01 * \sum_i SF_i = 0.91 + 0.01 * 13.63 = 1.0463$$

Thanks to the results in equations E and EAF, we are now able to estimate the value of the effort, defined in equation Effort which has a lower bound:

$$\text{Effort} = A * EAF * \text{KSLOC}^{E} = 2.94 * 0.76372 * 12.098^{1.0463} = 30.487 \text{ PM} \sim 30 \text{ PM}$$

And an upper bound:

$$\text{Effort} = A * EAF * \text{KSLOC}^{E} = 2.94 * 0.76372 * 16.569^{1.0463} = 42.367 \text{ PM} \sim 42 \text{ PM}$$

2.2.4 Schedule estimation

Now, thanks to the previous results, it is possible to evaluate the time to develop (Duration). Time to develop is the calendar time in months that goes from the determination of the product's requirements to the completion of an acceptance activity certifying that the product satisfies its requirements.
The time to develop is given by this formula:

Duration = $3.67 * Effort^{F}$

Exponent F is defined as follows:

F = 0.28 * 0.2 * (E-B) = 0.28 + 0.2 * (1.0463-0.91) = 0.30726
Effort = 30.487 PM
Duration = $3.67 * (30.487)^{0.30726}$ = 10.49 months

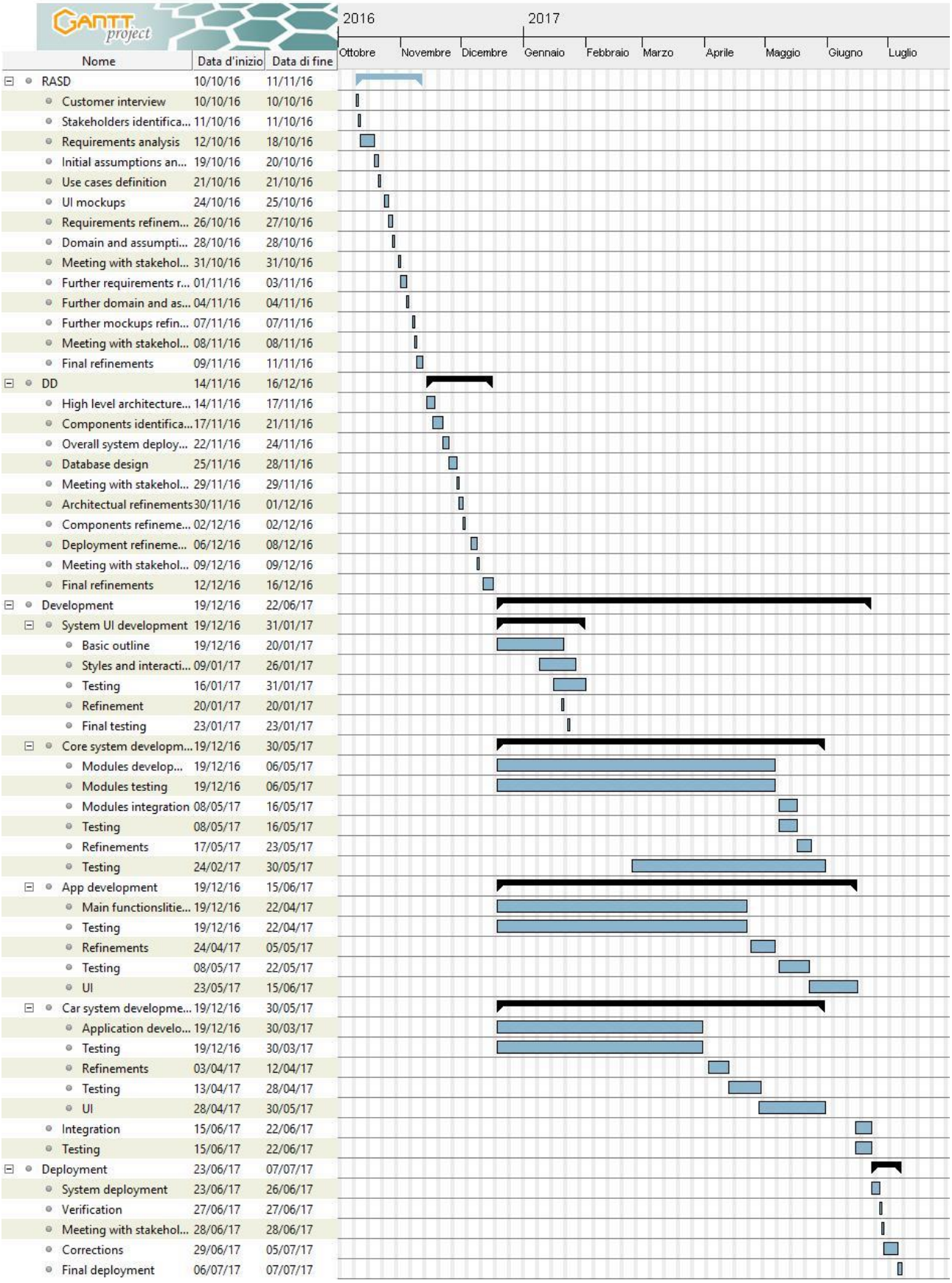
F = 0.28 * 0.2 * (E-B) = 0.28 + 0.2 * (1.0463-0.91) = 0.30726
Effort = 42.367 PM
Duration = $3.67 * (42.367)^{0.30726}$ = 11.60 months

3 Schedule

This chapter proposes a high level schedule comprising all the main phases being part of the project.

Four primary topics are identified:

- RASD: taking care of all the preliminary process of requirement analysis and specification definition
- DD: taking into account the need to find the most suitable design implementations
- Development: the core activity brought on in parallel on all the development edges, from the app to the core system. Testing results in being a major priority during all the development phase
- Deployment: the last phase, consisting in the real platform deployment. It's the last step of the process and it concludes with the service being active and ready to be used

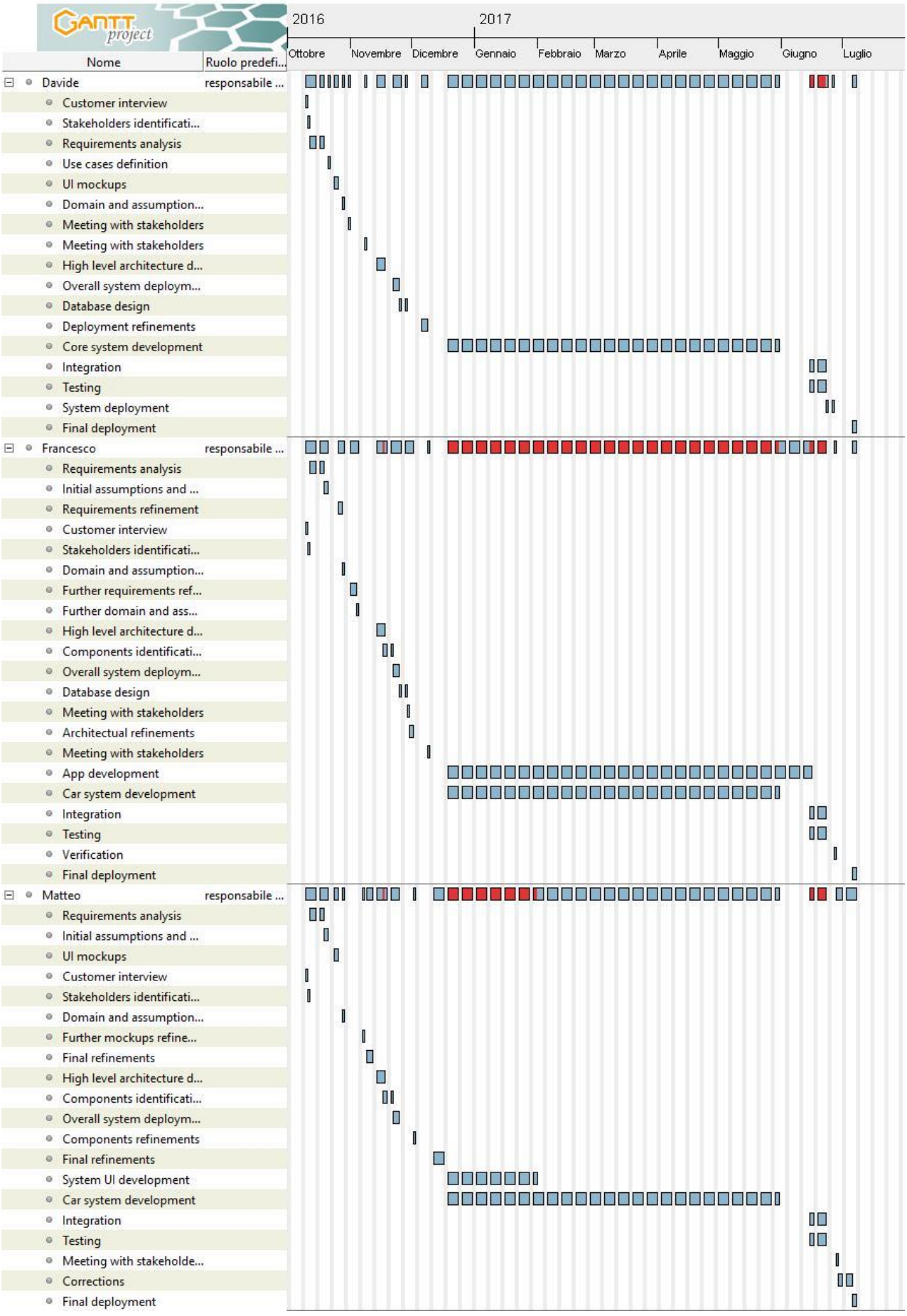


Nome	Data d'inizio	Data di fine
RASD	10/10/16	11/11/16
Customer interview	10/10/16	10/10/16
Stakeholders identifica...	11/10/16	11/10/16
Requirements analysis	12/10/16	18/10/16
Initial assumptions an...	19/10/16	20/10/16
Use cases definition	21/10/16	21/10/16
UI mockups	24/10/16	25/10/16
Requirements refinem...	26/10/16	27/10/16
Domain and assumpti...	28/10/16	28/10/16
Meeting with stakehol...	31/10/16	31/10/16
Further requirements r...	01/11/16	03/11/16
Further domain and as...	04/11/16	04/11/16
Further mockups refin...	07/11/16	07/11/16
Meeting with stakehol...	08/11/16	08/11/16
Final refinements	09/11/16	11/11/16
DD	14/11/16	16/12/16
High level architecture...	14/11/16	17/11/16
Components identifica...	17/11/16	21/11/16
Overall system deploy...	22/11/16	24/11/16
Database design	25/11/16	28/11/16
Meeting with stakehol...	29/11/16	29/11/16
Architectual refinements	30/11/16	01/12/16
Components refineme...	02/12/16	02/12/16
Deployment refineme...	06/12/16	08/12/16
Meeting with stakehol...	09/12/16	09/12/16
Final refinements	12/12/16	16/12/16
Development	19/12/16	22/06/17
System UI development	19/12/16	31/01/17
Basic outline	19/12/16	20/01/17
Styles and interacti...	09/01/17	26/01/17
Testing	16/01/17	31/01/17
Refinement	20/01/17	20/01/17
Final testing	23/01/17	23/01/17
Core system developm...	19/12/16	30/05/17
Modules develop...	19/12/16	06/05/17
Modules testing	19/12/16	06/05/17
Modules integration	08/05/17	16/05/17
Testing	08/05/17	16/05/17
Refinements	17/05/17	23/05/17
Testing	24/02/17	30/05/17
App development	19/12/16	15/06/17
Main functionslitie...	19/12/16	22/04/17
Testing	19/12/16	22/04/17
Refinements	24/04/17	05/05/17
Testing	08/05/17	22/05/17
UI	23/05/17	15/06/17
Car system developme...	19/12/16	30/05/17
Application develo...	19/12/16	30/03/17
Testing	19/12/16	30/03/17
Refinements	03/04/17	12/04/17
Testing	13/04/17	28/04/17
UI	28/04/17	30/05/17
Integration	15/06/17	22/06/17
Testing	15/06/17	22/06/17
Deployment	23/06/17	07/07/17
System deployment	23/06/17	26/06/17
Verification	27/06/17	27/06/17
Meeting with stakehol...	28/06/17	28/06/17
Corrections	29/06/17	05/07/17
Final deployment	06/07/17	07/07/17

4 Resource allocation

Purpose of this chapter is to outline the effort spent by each member of the group in finding the best possible requirements, design solutions and algorithms during the whole project creation cycle.

The effort in reported in terms of the three members of the group, each related to the tasks addressed during the process.


GANTT project
2016
2017
Ottobre
Novembre
Dicembre
Gennaio
Febbraio
Marzo
Aprile
Maggio
Giugno
Luglio
Nome
Ruolo predefi...
Davide
responsabile ...
Customer interview
Stakeholders identificati...
Requirements analysis
Use cases definition
UI mockups
Domain and assumption...
Meeting with stakeholders
Meeting with stakeholders
High level architecture d...
Overall system deploym...
Database design
Deployment refinements
Core system development
Integration
Testing
System deployment
Final deployment
Francesco
responsabile ...
Requirements analysis
Initial assumptions and ...
Requirements refinement
Customer interview
Stakeholders identificati...
Domain and assumption...
Further requirements ref...
Further domain and ass...
High level architecture d...
Components identificati...
Overall system deploym...
Database design
Meeting with stakeholders
Architectual refinements
Meeting with stakeholders
App development
Car system development
Integration
Testing
Verification
Final deployment
Matteo
responsabile ...
Requirements analysis
Initial assumptions and ...
UI mockups
Customer interview
Stakeholders identificati...
Domain and assumption...
Further mockups refine...
Final refinements
High level architecture d...
Components identificati...
Overall system deploym...
Components refinements
Final refinements
System UI development
Car system development
Integration
Testing
Meeting with stakeholde...
Corrections
Final deployment

5 Risk management

During the analysis phase of the project, a few risks have been identified as potential dangers for the project.

First, due to the need for urban permissions to build power charge stations and the high cost needed for their construction, overall costs could raise rapidly, especially if PowerEnJoy wants the power stations to be fairly distributed.

Second, due to the unpredictable nature of the customers, it might happen to be in need to go take back in the city a car that was brought of the prefixed boundaries. This could cause an increase in costs and might also have legal consequences.

Third, battery degradation over time is definitely a factor that must be taken into account. Due to the volatile nature of the good, battery replacements may result in a substantial cost voice.

A solution to these three kinds of risk is not immediately identifiable. The suggestion is to take into account the need for an extra budget, exceeding the one allocated for common manutention operations, to be used in case such emergencies should occur.

Finally, time constraints might be not respected due to any problem in the development chain. This could eventually cause delays in the implementation of the project.

To manage this last eventuality, the estimation for the hours needed to conclude the project have been overestimated by a 30%.

6 Revision

6.1.1 Team work

Here is reported a compact table showing how the work was brought on by all the members of the group.

Document work finished on the xth January 2017.

effort spent by each group member - assignment 4

francesco				matteo				davide			
ora inizio	ora fine	dettaglio lavoro	totale ore	ora inizio	ora fine	dettaglio lavoro	totale ore	ora inizio	ora fine	dettaglio lavoro	totale ore
17/01 13.00	06/01 21.00	function point + cocomo II	8.00.00	17/1/17 20.00	17/1/17 22.30	document elaboration	2.30.00	2017-01-18 16.3	18/01 19.00	document structure + chapter 2 integration	2.30.00
19/01 19.00	19/01 21.00	fix cocomo II	2.00.00	19/1/17 13.30	19/1/17 17.30	functional points	4.00.00	19/01 19.30	19/01 22.00	Gantt diagram	2.30.00
22/01 10.00	08/01 12.00	final corrections and checks	2.00.00	21/1/17 16.30	21/1/17 18.30	document fixes and correction	2.00.00	21/01 13.00	21/01 14.30	Gantt correction and integration	1.30.00
			0.00.00				0.00.00	21/01 19.00	21/01 20.00	Ducument headers descriptions and content table	1.00.00